**PCT** WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 17/30 — A1

(11) International Publication Number: WO 00/58865

(43) International Publication Date: 5 October 2000 (05.10.00)

(21) International Application Number: PCT/IB00/00188

(22) International Filing Date: 22 February 2000 (22.02.00)

(30) Priority Data:
99106610.1 31 March 1999 (31.03.99) EP

(71) Applicant *(for all designated States except US)*: INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; New Orchard Road, Armonk, NY 10504 (US).

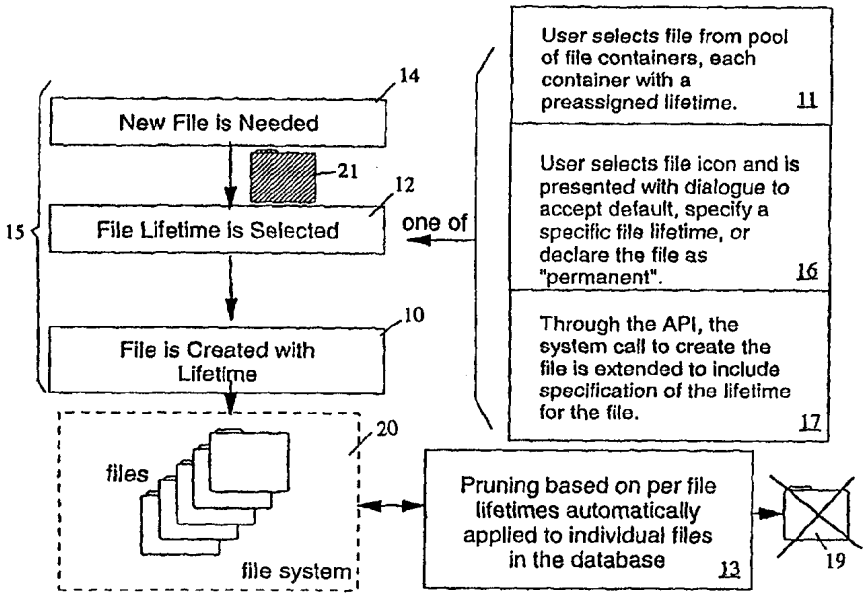
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: FERIDUN, Metin [US/CH]; Sonnenbergstrasse 7, CH–8800 Thalwil (CH). RUDIN, Harry [CH/CH]; Vordere Bergstrasse 1, CH–8942 Oberrieden (CH).

(74) Agent: HEUSCH, Christian; International Business Machines Corporation, Saeumerstrasse 4, CH–8803 Rueschlikon (CH).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published**
*With international search report.*

(54) Title: AUTOMATED FILE PRUNING




(57) Abstract

Scheme for adding an element (21) to a database system (20) which comprises a memory with limited storage capacity. The element (21) comprises data and metadata. An individual lifetime is defined for the element (21) in the metadata, and then the data and said metadata are stored in the memory. The metadata are examined from time–to–time to determine whether the lifetime of an element expired. If this is the case the respective element (19) is automatically removed.

*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **AL** | Albania | **ES** | Spain | **LS** | Lesotho | **SI** | Slovenia |
| **AM** | Armenia | **FI** | Finland | **LT** | Lithuania | **SK** | Slovakia |
| **AT** | Austria | **FR** | France | **LU** | Luxembourg | **SN** | Senegal |
| **AU** | Australia | **GA** | Gabon | **LV** | Latvia | **SZ** | Swaziland |
| **AZ** | Azerbaijan | **GB** | United Kingdom | **MC** | Monaco | **TD** | Chad |
| **BA** | Bosnia and Herzegovina | **GE** | Georgia | **MD** | Republic of Moldova | **TG** | Togo |
| **BB** | Barbados | **GH** | Ghana | **MG** | Madagascar | **TJ** | Tajikistan |
| **BE** | Belgium | **GN** | Guinea | **MK** | The former Yugoslav Republic of Macedonia | **TM** | Turkmenistan |
| **BF** | Burkina Faso | **GR** | Greece | | | **TR** | Turkey |
| **BG** | Bulgaria | **HU** | Hungary | **ML** | Mali | **TT** | Trinidad and Tobago |
| **BJ** | Benin | **IE** | Ireland | **MN** | Mongolia | **UA** | Ukraine |
| **BR** | Brazil | **IL** | Israel | **MR** | Mauritania | **UG** | Uganda |
| **BY** | Belarus | **IS** | Iceland | **MW** | Malawi | **US** | United States of America |
| **CA** | Canada | **IT** | Italy | **MX** | Mexico | **UZ** | Uzbekistan |
| **CF** | Central African Republic | **JP** | Japan | **NE** | Niger | **VN** | Viet Nam |
| **CG** | Congo | **KE** | Kenya | **NL** | Netherlands | **YU** | Yugoslavia |
| **CH** | Switzerland | **KG** | Kyrgyzstan | **NO** | Norway | **ZW** | Zimbabwe |
| **CI** | Côte d'Ivoire | **KP** | Democratic People's Republic of Korea | **NZ** | New Zealand | | |
| **CM** | Cameroon | | | **PL** | Poland | | |
| **CN** | China | **KR** | Republic of Korea | **PT** | Portugal | | |
| **CU** | Cuba | **KZ** | Kazakstan | **RO** | Romania | | |
| **CZ** | Czech Republic | **LC** | Saint Lucia | **RU** | Russian Federation | | |
| **DE** | Germany | **LI** | Liechtenstein | **SD** | Sudan | | |
| **DK** | Denmark | **LK** | Sri Lanka | **SE** | Sweden | | |
| **EE** | Estonia | **LR** | Liberia | **SG** | Singapore | | |

# AUTOMATED FILE PRUNING

## TECHNICAL FIELD

The invention concerns a scheme for automated file pruning and an automated file pruning facility for use in computer systems and networks.

## BACKGROUND OF THE INVENTION

A typical computer system comprises a central processing unit, memory, peripheral devices such as data input/output devices and storage media such as floppy disks or hard disks. The devices communicate with each other via a computer operating system. Computer operating systems include several different operational modules. One such module might be a file manager. Client data stored on a storage medium is organized as a file system having an associated format, and the file manager is employed to maintain the available file systems and control access to them. The file manager provides a set of system calls in the form of Application Programmer Interfaces (APIs) to allow clients access to the file system and files, to carry out operations such as file creation, deletion, opening for read or write, and so on.

The file system stores, organizes and describes client data. The client data is stored in files. The files are arranged in each file system in a particular format. Each file system maintains its organization and some sort of description information, herein referred to as metadata, in format specific structures. Examples of such formats are HFS, MS-DOS FAT, ProDOS, High Sierra, ISO 9660, NTFS, and so on. The term format encompasses both physical disk format and network server access protocols.

To read and write data from and to the file system, the file manager must be able to recognize the format of the file system.

It is a well known disadvantage of current file systems that as the number of files increases the handling becomes more and more difficult. Fast access or retrieval and a clear overview of the whole file system are crucial for its daily use.

Despite the growing capacity of hard disks, the memory of a computer system remains in short demand, specially in light of increasing use of applications that require larger amounts

of storage, such as multimedia applications. The presence of too many files not only consumes unnecessary memory capacity but - because of the time it takes to find and access the files stored - substantially worsens system performance in terms of response time.

Each computer user has a huge variety of different kinds of files. In current file systems all these files are treated the same. If the computer system is used in a commercial environment, there are usually so-called record retention rules which define the different kinds of documents, where and how they have to be stored in memory, and - most important - how long the respective documents have to be archived. Current file systems do not support a user in adhering to such record retention rules.

Pruning facilities exist for removing files which have exceeded a common age but these are uniformly applied over the entire file system. For example, in a UNIX system, optional, periodically running software can be programmed to remove files in a list of folders (or directories) which are older than a set number of days. It is a disadvantage of such a global approach that all files are treated the same, no matter how important they are. Furthermore, systematic removal of files as described above is not a mandatory part of the operating system but a local customized option: with the increase use of unadministered personal workstations or laptops, the growth of files and the subsequent disk memory requirements can be substantial.

It is an object of the present invention to provide a scheme which makes efficient use of the limited memory resources of a computer system.

It is an object of the present invention to provide a scheme which allows archiving of files according to predefined rules, such as record retention rules, for example.

## SUMMARY OF THE INVENTION

The objectives of the present invention have been accomplished by the method, computer program products, computer program elements, and system as claimed.

Advantages of the present invention are addressed in connection with the detailed description or are apparent from the description.

## DESCRIPTION OF THE DRAWINGS

The invention is described in detail below with reference to the following schematic drawings.

**FIG. 1** is an illustration of an exemplary embodiment of the present invention.

## DESCRIPTION OF PREFERRED EMBODIMENTS

In the following, the basic concept of the present invention is described. Before addressing different embodiments, the relevant terms and expressions are defined and explained.

The expression "element" is herein used to describe an element of a database. Examples of elements are: files (e.g., computer or machine readable files), partial information to be added to an existing file, an image from an image repository, for example, a multimedia object (e.g., an avi or mpeg movie).

The word "database" is herein used to describe any collection, library, or repository of information, such as a file database, or an image repository. Examples of databases are a DOS or Windows file directory. A database system is the system which houses the database. A database system and/or a database can be distributed.

An element content is herein referred to as data. An element might for example comprise client data, data generated by an application program, computer code for execution by a processor, and so forth.

The elements are arranged in each database system in a particular format and each database system maintains its organization and some sort of description information, herein referred to as metadata, in format specific structures. The metadata may contain the following exemplary information: file attributes, file name, date of creation, date of last revision, users with access permission, file size, file type (e.g., Lotus WordPro), etc. The metadata can be kept in the same memory as the elements' data or they can be kept in a separate memory, e.g., a cache memory. The metadata are like a card catalog that is found in the library, with the library being analogous to the database system.

In general, a database system enables the user to store, manage, share and reuse information about the elements which are stored in it. The repository enables the user to store more than just the elements' data. For example, the metadata stored in the database system may comprise information about the development of applications; including descriptions of data, programs and system objects. It may also include information about relationships among data, programs and system objects; as well as the semantics and use of the information.

The basic idea of the present scheme is to provide an automated file pruning facility which automatically removes elements (e.g. files) no longer needed from a database system. There are three main components: one for element (file) generation, one for element (file) representation, and one for element (file) disposal. These three components are described in connection with a file database comprising files.

**File Generation:** During file generation, e.g. when creating a file or adding a file to a database, a field is added to the file's metadata (e.g. to the file descriptor). This new field, which specifies the lifetime of the file, has to be initially set at the time a new file is generated or before the file is stored in the file database.

When creating a file 21 (steps 14, 12, and 10 in Figure 1) or adding a file 21 to a database 20 which comprises a memory with limited storage capacity, an individual lifetime is defined in the metadata (step 12) of this element. Then, the element's data and its metadata are stored in the memory. Note that an individual lifetime is assigned to each element.

**File Representation or Marking:** When a file 21 is first created the user has to select a lifetime (step 12). This can be done by asking the user to accept a default file lifetime, to

specify a specific file lifetime, or to declare the file as "permanent", as illustrated in box 16. Likewise, the user might also be prompted by the system to categorize the file 21 using a predefined categorization scheme (box 11). If the file 21 is categorized in a group of "legal documents", the default lifetime used for all legal documents (e.g. 20 years from data of creation) is assigned to the respective file 21. If the user categorizes the file 21 into a directory with private documents, a shorter period might be assigned, just to give an example.

Another alternative which is addressed later is illustrated in box 17 of Figure 1.

The lifetime can be defined by a user or an application program. Information concerning record retention rules might be provided to the user prior to defining the lifetime of a particular file.

**File Disposal:** When the file lifetime limit is reached, the file 19 will be erased, as indicated on the right hand side of Figure 1. This could be done by code scanning file descriptions in search for files which have reached their end-of-life (step 13). This code be run automatically at a specified frequency. An optional feature could be employed that asks the user to approve the list of files to be erased.

The metadata are examined from time-to-time to determine whether the lifetime of an element expired such that an element can be removed from said memory if its lifetime expired. Likewise, a user can be prompted if the lifetime of an element is about to expire or after it expired but before it is deleted by the system. In this case, the user has the option to chose whether the element is to be removed from said memory, or whether to change the element's lifetime.

**Implementation:** In the following a first embodiment is given. This embodiment is based on an object-oriented framework.

In a file database system using an object-oriented approach where each file is represented as an object class, the normal class "file" can be extended (e.g., through the inheritance mechanism in object-oriented programming) to a new class. This new class can be called "limited lifetime" class for example. The "limited lifetime" class comprises all those files which the user would like to have automatically removed at a later date. When a file of type

"limited lifetime" is generated, the user would be asked to accept a default lifetime or to specify a particular file lifetime. Together, the current date and the lifetime specify a kill time for the file at which time it can be deleted. When the file is created, it is registered with the database or file manager as a "limited lifetime" type. The database or file manager will periodically scan its list of files with the type of "limited lifetime" and remove any whose kill time had expired.

Another embodiment is described in connection with Figure 1. This Figure shows a schematic representation of a file system 20 with several files. It furthermore shows a sequence of steps 15 for adding a new file 21, and a process for pruning (step 13) of the file system 20. According to the present embodiment, there are three steps that are carried out if a new file 21 is needed (step 14). A lifetime has to be selected (step 12), according to the present invention. Three options (box 11, box 16, and box 17) are offered for selection of the lifetime.

As indicated in box 11, there is a pool of file containers (e.g. represented by icons on a display). Each container has a preassigned lifetime. There might be a legal container that has a lifetime of 20 years, a temporary container with a lifetime of 10 days, and an accounting container with a lifetime of 10 years, for example. If the user wants to create a new file 21 that contains legal information, or that belongs in the legal category, he selects the respective legal container during step 12. The container can be selected by clicking on it (if a graphical user interface is employed), by using hot keys, or the like. By doing so the legal container's preassigned lifetime is assigned to the new file 21.

A different approach for selection of the lifetime is illustrated in box 16. The user may select a file, a file icon, or use an application program to create a new file 21. In this context he is presented with a dialogue to accept a default lifetime, specify a specific file lifetime, or to declare this file as permanent. This can be done by using appropriate icons, hot keys, or other well known means.

The processes depicted in boxes 11 and 16 are well suited for systems which help the user to observe record retention rules. The approach of box 11 is particularly well suited because the user does not need to worry about record retention rules at all. He just needs to put his newly created files in the right category. The rest is done by the system.

Another alternative is illustrated in box 17 where the system call to create a file 21 is extended to include the specification of the desired lifetime for the file 21. This can be done through the application programming interface (API).

Once the lifetime is selected, the actual file is created (step 10) and the lifetime is stored in the file's metadata. It is obvious that the sequence of steps can be changed. It is conceivable, for example, that a file 21 is created if needed by using an application software (e.g. a text processing software). When saving the newly created file to disk, or when adding it to the file system 20, the user may be asked to select a lifetime.

From time-to-time, either on a regular basis or at random intervals, the file system is automatically pruned. This pruning process is indicated by box 13. If during the pruning process a file 19 is discovered whose lifetime expired, then the respective file is discarded. As a optional safety net, the user might be prompted prior to this. Instead of prompting the user, all the files that are about to be deleted can be put in a waste basket. Here the files remain for a certain period of time. This gives the user an opportunity to intervene. If he does not remove the files from the waste basket, then the files are finally deleted.

A system in accordance with the present invention can also offer a feature which allows the user to act upon an element (e.g. a file) that is stored in a database (e.g. a file system). For this purpose the metadata of the respective element is fetched and displayed. Then, the user can act upon the element. Typical actions are: accessing, retrieving, opening, displaying, moving, or copying the element. The user can also changing the element's lifetime or he can link the lifetime with the lifetime of another element. The linking can be done by a drag-and-drop operation where the element is moved into a container with a desired preassigned lifetime, for example.

The present invention can be realized in hardware, software, or a combination of hardware and software. Any kind of computer system or other apparatus adapted for carrying out the methods described herein is suited. A typical combination of hardware and software could be a general purpose computer system with a computer program that, when being loaded and executed, controls the computer system such that it carries out the methods described herein. The present invention can also be embedded in a computer program product, which comprises

all the features enabling the implementation of the methods described herein, and which - when loaded in a computer system - is able to carry out these methods.

The present invention can also be used in systems with parallel data sharing access to files residing on network attached shared disks.

A special viewer can be used to review the lifetime of the files in a database. The viewer might offer a tool that organizes the files on a screen or within a window by their lifetime, for example. The viewer can be used to act upon the files and/or their lifetimes. Let us assume that one company decides to take legal action against another company. In such a situation all the documents that are of relevance are to be produced. It is important to avoid that these documents are deleted. All relevant documents can thus be displayed by the viewer and the user can act upon them. He might for example assign a new lifetime to each such document to ensure that the are kept for another five years.

CLAIMS

1. Method for adding an element (21) to a database system (20) comprising a memory with limited storage capacity, whereby said element (21) comprises data and metadata, comprising the steps:

   - defining an individual lifetime for said element (21) in said metadata,
   - storing said data and said metadata in said memory.

2. The method of claim 1, whereby said metadata are examined from time-to-time to determine whether said lifetime of an element expired.

3. The method of claim 2, whereby an element (19) is removed from said memory if its lifetime expired.

4. The method of claim 2, whereby a user is prompted if said lifetime of an element is about to expire or expired.

5. The method of claim 4, whereby said user can chose whether said element is to be removed from said memory, or whether to change said element's lifetime.

6. The method of claim 1, whereby said lifetime is defined using a default setting.

7. The method of claim 6, whereby said default setting depends on the element type, or the data of said element.

8. The method of claim 1, whereby said lifetime is defined by a user or an application program.

9. The method of claim 8, whereby information concerning record retention rules is provided if said user defines said lifetime.

10. The method of claim 1, whereby said element is a file (21) and said database system is as file repository or file directory.

11. The method of claim 1, whereby said individual lifetime for said element is defined by displaying a selection of element containers, each container with a preassigned lifetime, such that the individual lifetime can be selected by selecting one of said containers.

12. Method for acting upon an element (21) comprising data and metadata kept in a database system's memory (20) with limited storage capacity whereby said metadata comprises a lifetime for said element (21), comprising the steps:

- fetching said metadata,
- displaying said lifetime for said element, and
- acting upon said element (21).

13. The method of claim 12, whereby acting upon said element refers to one of the following actions: accessing said element, retrieving said element, opening said element, displaying said element, moving said element, or copying said element.

14. The method of claim 12, whereby acting upon said element refers to one of the following actions: changing said lifetime, or linking said lifetime with the lifetime of another element.

15. The method of claim 12, whereby said element is acted upon by changing its lifetime.

16. The method of claim 15, whereby said lifetime is changed by moving said element into a container with a desired preassigned lifetime.

17. A computer program product comprising a computer readable medium, having thereon:

computer program code means, when said program is loaded, for adding a new element (21) to a database system (20) which comprises a memory with limited storage, capacity, said database system (20) containing multiple elements which comprise data and metadata, execute procedure to

- define an individual lifetime in the metadata of the new element (21),

- store the data and metadata of said new element (21) in said memory.

18. A computer program element comprising:

computer program code means for adding a new element (21) to a database system (20) which comprises a memory with limited storage, capacity, said database system (20) containing multiple elements which comprise data and metadata, execute procedure to

- define an individual lifetime in the metadata of the new element (21),

- store the data and metadata of said new element in said memory.

19. The computer program product of claim 17, or the computer program element of claim 18, wherein said metadata are examined from time-to-time to determine whether said lifetime of an element expired.

20. The computer program product of claim 17, or the computer program element of claim 18, wherein an element is removed from said memory if its lifetime expired.

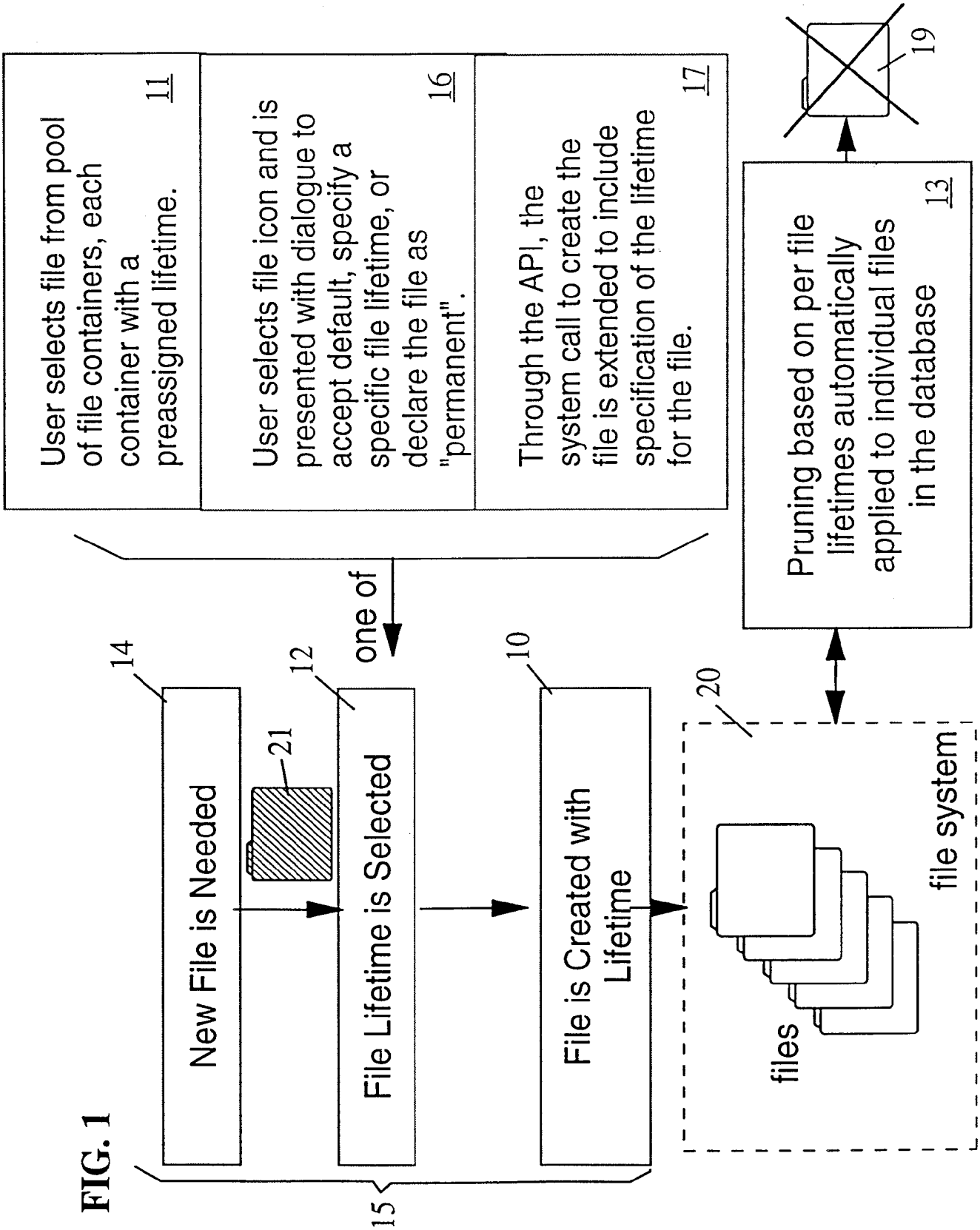

User selects file from pool of file containers, each container with a preassigned lifetime. 11
User selects file icon and is presented with dialogue to accept default, specify a specific file lifetime, or declare the file as "permanent". 16
Through the API, the system call to create the file is extended to include specification of the lifetime for the file. 17
one of
14
New File is Needed
21
12
File Lifetime is Selected
10
File is Created with Lifetime
15
20
files
file system
Pruning based on per file lifetimes automatically applied to individual files in the database 13
19

FIG. 1

# INTERNATIONAL SEARCH REPORT

Inte ional Application No
PCT/IB 00/00188

**A. CLASSIFICATION OF SUBJECT MATTER**
IPC 7 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category ° | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | US 5 689 699 A (REDDY HARI N ET AL)<br>18 November 1997 (1997-11-18)<br>abstract<br>column 1, line 1 -column 4, line 45<br>--- | 1-5,<br>10-20 |
| X | EP 0 323 025 A (IBM)<br>5 July 1989 (1989-07-05)<br>abstract<br>page 1, line 1 -page 2, line 11<br>page 4, line 41 -page 6, line 49<br>figures 11,12A,12B<br>---<br>-/-- | 1-3,6-9,<br>17-20 |

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 12 April 2000 | 22/05/2000 |

| Name and mailing address of the ISA | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL – 2280 HV Rijswijk<br>Tel. (+31–70) 340–2040, Tx. 31 651 epo nl,<br>Fax: (+31–70) 340–3016 | Corcoran, P |

3

Form PCT/ISA/210 (second sheet) (July 1992)

# INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/IB 00/00188

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

| Category ° | Citation of document, with indication,where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | SHARICK P: "Expiration dates and retention periods. Keeping your disks clean" VAX PROFESSIONAL, JUNE 1988, USA, vol. 10, no. 3, pages 15-17, XP000885698 ISSN: 8750-9628 "Deleting expired files" page 17 "Selecting a retention period" page 16 "What is a retention period" page 15 -page 16; figures 1,2 --- | 4-7, 11-16 |
| A | ANONYMOUS: "Deletion Confirmation for Hierarchical/ Related Entities." IBM TECHNICAL DISCLOSURE BULLETIN, vol. 34, no. 8, 1 January 1992 (1992-01-01), pages 376-377, XP002135154 New York, US the whole document --- | 4,5 |
| A | ANONYMOUS: "Generic Cascade Menu to Create New Object Instances" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 37, no. 9, 1 September 1994 (1994-09-01), pages 347-350, XP002135155 New York, US the whole document ----- | 11,16 |

3

# INTERNATIONAL SEARCH REPORT

Information on patent family members

Int ional Application No
PCT/IB 00/00188

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 5689699 A | 18-11-1997 | NONE | |
| EP 0323025 A | 05-07-1989 | US 5107419 A | 21-04-1992 |
| | | CA 1306069 A | 04-08-1992 |
| | | JP 1173159 A | 07-07-1989 |
| | | JP 2053152 C | 10-05-1996 |
| | | JP 6005510 B | 19-01-1994 |